

King Fahd University of Petroleum & Minerals

College of Computer Sciences and Engineering

Information and Computer Science Department

ICS 202: Data Structures

**Project**

This project consists of two phases:

**Phase I (20% of the Project): Due (Sunday April 8, 2018 at midnight)**

Write a java program that takes as input a colored image (represented by the image filename) and produces the following:

1. A gray-scale image from the colored image applying the following function to each pixel of the image

0.2125 \* redComponent + 0.7154 \* greenComponent + 0.0721 \* blueComponent

1. Add white noise to the resulting gray-scale image. The noise amount noiseAmt is computed according to the formula

noiseAmt = (int)(255\*(Math.random()-0.5))

for each pixel. Then, it is added to the pixel value, such that the new pixel value is greater than or equal to zero and less than or equal to 255. This noise amount is computed for each pixel, hence it is not a fixed value.

**Requirements:**

I. One submission is required from each group.

II. A README file should be included that explains how to compile and run the code and/or specify any other requirements related to running the code. Provide text files that contain sample input that the grader can use in order to cut and paste while running the program.

**Grading:(out of 100)**

1. Simplicity of compiling and running the program 20 pts.

2. Programming: 80 pts.

**Phase 2 (80% of the project): Due (Sunday April 22, 2018 at midnight)**

1. Write a java program that takes as input a colored image (represented by the image filename) and an integer, , and produces as output the same image segmented into regions using
2. The minimum spanning tree-based method (MST) using the 4-neighbouhood and the 8-neighbourhood image-to-graph transformation.
3. The recursive minimum spanning tree-based method (RMST) using the 4-neighbouhood and the 8-neighbourhood image-to-graph transformation.
4. Run your implementations in part A on the given images and show the respective results.
5. Analyze the algorithm in A based on the following:
6. Compare the results of 4-neighbourhood RMST and 4-neighbourhood MST when the number of regions specified is 2%, 5% and 15% of the image size.
7. Compare the running times of MST-based vs. RMST-based segmentation.
8. Add white noise to the images in part C.1 and repeat the comparison of part C.1.
9. Repeat C.1, C.2 and C.3 using 8-neighbourhood implementations instead of 4-neighbourhood. Do you notice any difference between using the 4-neighborhood implementation and the 8-neighborhood implementation?

**Requirements:**

I. One submission is required from each group.

II. A README file should be included that explains how to compile and run the code and/or specify any other requirements related to running the code. Provide text files that contain sample input that the grader can use in order to cut and paste while running the program.

III. An assignment report should be included which contains the following information:

III.i: Team Information: Names and IDs.

III.ii: Algorithms and data structures you used and justification for your design choices.

III.iii: Your analysis of the algorithm as required in Part C. Include snapshots and/or images that were generated and are used in your analysis.

III.iv: Who did what in the programming assignment (Reported for both phases).

**References:**

1. Project Background Handout.
2. O.J. Morris, M. de J. Lee and A.G. Constantinides. *A Unified method for segmentation and edge detection using Graph Theory*. In the proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 1986.

**Grading:(out of 100)**

1. Simplicity of compiling and running the program 10 pts.

2. Programming: 60 pts.

3. Assignment Report: 30 pts.